HP StorageWorks
Clustered File System 3.0
Command Line
reference guide


392372–001

Part number: 392372–001
First edition: May 2005

Legal and notice information

Clustered File System Command Line Reference Guide

Contents

HP Technical Support

Telephone numbers for worldwide technical support are listed on the following HP web site: http://www.hp.com/support. From this web site, select the country of origin. For example, the North American technical support number is 800-633-3600.

NOTE: For continuous quality improvement, calls may be recorded or monitored.

Be sure to have the following information available before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

HP Storage Web Site

The HP web site has the latest information on this product, as well as the latest drivers. Access the storage site at: http://www.hp.com/country/us/eng/prodserv/storage.html. From this web site, select the appropriate product or solution.

HP NAS Services Web Site

The HP NAS Services site allows you to choose from convenient HP Care Pack Services packages or implement a custom support solution delivered by HP ProLiant Storage Server specialists and/or our certified service partners. For more information see us at http://www.hp.com/hps/storage/ns_nas.html.

1

HP Clustered File System Commands

HP Clustered File System includes several commands that can be helpful for administrators managing a HP Clustered File System cluster. Other HP Clustered File System commands provide diagnostic information and should be used only under the direction of HP personnel. HP Clustered File System also includes commands that are used internally and should not be run directly.

The HP Clustered File System commands are located in the following directories:

- */opt/hpcfs/bin*
- */opt/hpcfs/sbin*
- */opt/hpcfs/tools*
- */etc/init.d*

clusterpulse – ClusterPulse daemon

Synopsis

/opt/hpcfs/sbin/clusterpulse

Description

The ClusterPulse daemon monitors the cluster, controls failover of virtual hosts and devices, handles communications with the Management Console, and manages monitors and event notification.

This daemon is used internally by HP Clustered File System and should never be run directly.

destroypsfs – destroy a PSFS filesystem

Synopsis

/opt/hpcfs/sbin/destroypsfs

Description

This command removes the specified PSFS filesystem from the volume on which it is located.

diskupdate – reattempt to access a disk

Synopsis

/opt/hpcfs/tools/diskupdate

Description

This command should be run only at the request of HP personnel.

dlm – Distributed Lock Manager daemon

Synopsis

/opt/hpcfs/sbin/dlm

Description

The Distributed Lock Manager daemon provides a locking mechanism to coordinate server access to shared resources in the cluster.

This daemon is used internally by HP Clustered File System and should never be run directly.

dlmdebug – debug DLM problems

Synopsis/

/opt/hpcfs/tools/dlmdebug

Description

This utility should be run only at the request of HP personnel.

emulex/dfc, emulex/lpedit – Emulex utilities

Synopsis

/opt/hpcfs/tools/emulex/dfc
/opt/hpcfs/tools/emulex/lpedit

Description

These Emulex utilities can be used to view information about Emulex HBAs and to perform tasks such as upgrading firmware. See your Emulex documentation for more information.

fsprobe – report filesystem information

Synopsis

/opt/hpcfs/tools/fsprobe

Description

This command should be run only at the request of HP personnel.

gcstat – print grpcommd statistics

Synopsis

/opt/hpcfs/tools/gcstat

Description

This command should be run only at the request of HP personnel.

get_fenceidentity – get fencing information

Synopsis

/opt/hpcfs/sbin/get_fenceidentity

Description

The **get_fenceidentity** utility retrieves the fence identification information for the system on which it is run. The utility is used internally during HP Clustered File System configuration and should not be run manually.

grpcommd – cluster-wide communications daemon

Synopsis

/opt/hpcfs/sbin/grpcommd

Description

The grpcommd daemon manages HP Clustered File System group communications across the cluster. It is used internally by HP Clustered File System and should never be run directly.

hbaapidump – show LUN information

Synopsis

/opt/hpcfs/tools/hbaapidump

Description

This tool displays information about the LUNs seen by the HBA drivers and can be used to verify that the HBA libraries are working correctly.

lcl-dump – show information about LCL locks

Synopsis

/opt/hpcfs/tools/lcl-dump

Description

This command should be run only at the request of HP personnel.

log_collect – obtain log files

Synopsis

/opt/hpcfs/tools/log_collect

Description

This command is used internally by the **mxcollect** utility and should not be run directly.

mkpsfs – create a PSFS filesystem

Synopsis

/opt/hpcfs/sbin/mkpsfs [**-n** *<max-nodes>*] [**-l** *<label>*] *<device>* [*<size-in-blocks>*]

Description

The **mkpsfs** command creates a PSFS filesystem on the specified device, which must be imported into the cluster. PSFS filesystems use a block size of 4 KB.

device is a psd or psv device and is specified as follows:

- For a psd device partition, the device is specified as **/dev/psd/psd***XXX***p***YY*, where XXX is the drive number and YY is the partition number. For example, **/dev/psd/psd6p4** specifies partition 4 on disk psd6.
- For a non-partitioned psd device, the device is specified as **/dev/psd/psd***XXX*, where *XXX* is the drive number. For example, /**dev/psd/psd5**.
- For a psv device, the device is specified as **/dev/psv/psv***XXX*, where *XXX* is the volume number. For example, **/dev/psv/psv1**.

mkpsfs requires exclusive access to the device. If it cannot obtain exclusive access, it will exit with an “ebusy” message.

The arguments are as follows:

size-in-blocks
: The number of blocks on the device. If this value is omitted, **mkpsfs** will determine it automatically.

-n *max-nodes*

The maximum number of nodes that can be in the cluster.

-l *label*

The label to be applied to the filesystem.

mx – HP Clustered File System command-line interface

The **mx** utility provides a command-line interface for administering a cluster and monitoring its operation. See Chapter 2 for more information about the commands provided with this utility.

mxcheck – verify HP Clustered File System requirements

Synopsis

/opt/hpcfs/bin/mxcheck

Description

The **mxcheck** utility verifies that a server meets the configuration requirements for HP Clustered File System. It is run automatically each time HP Clustered File System is started. Output from the utility appears on the screen and is also written to the */var/hpcfs/mxcheck* directory. If you are experiencing errors, check the messages in this directory.

The **mxcheck** utility performs checks such as the following: operating system version and configuration; available physical memory and disk space; HBA driver versions; FibreChannel switch versions; network addresses. The utility also attempts to access gateways and FC switches.

mxcollect – collect configuration information

Synopsis

/opt/hpcfs/tools/mxcollect

Description

This script collects configuration information and log files from the cluster. The script is typically run under the direction of HP Technical Support.

mxconfig – configure the cluster

Synopsis

/opt/hpcfs/bin/mxconfig [**--import**] [**--help**]

Description

The **mxconfig** utility performs the initial configuration of HP Clustered File System. **mxconfig** invokes a dialog-based interface that allows you to install the HP Clustered File System license file, to select a cluster password and a Network Authentication Secret password, to select a fencing method and optionally configure the FibreChannel switches to be used in the cluster, and to select the SAN disk partitions to be used as HP Clustered File System membership partitions.

You can run **mxconfig** on one server and then export the resulting configuration to the other servers. You must be *root* to run this utility.

The options are:

--help
Display a usage message.

--import
Import a HP Clustered File System configuration from an existing cluster server to a new server.

See the *HP StorageWorks Clustered File System Setup Guide* for more information about **mxconfig**.

mxconsole – start the Management Console

Synopsis

/opt/hpcfs/bin/mxconsole

Description

The **mxconsole** command starts the Management Console, which is used to configure and monitor the cluster. The Management Console can be run from either a cluster server or a local machine outside the cluster. See the *HP StorageWorks Clustered File System Administration Guide* for information about using the Management Console.

mxfence – verify fencing module configuration

Synopsis

/opt/hpcfs/sbin/mxfence *<server hostname or IP address>*

Description

The **mxfence** utility can be used to verify that HP Clustered File System has the information needed to fence a server. This utility is intended for server-based fencing configurations where you must specify the hostname or IP address of the remote controller associated with the server. You can use **mxfence** to verify that HP Clustered File System has the correct information. The server must be up when you use the utility.

When you run **mxfence**, HP Clustered File System uses the hostname/IP address associated with the server to access the remote controller. The server is then either power-cycled or shut down in accordance with the method you selected when you configured the fencing module.

If the correct server is not fenced, the hostname or IP address specified for the associated remote controller is probably incorrect. Run **mxconfig** on the server with the incorrect information and then modify the hostname or IP address for the remote controller.

mxfencecfg – configure the fence agent

Synopsis

/opt/hpcfs/bin/mxfencecfg
mxfencecfg **-r**|**-s** **-v** *<vendor>*
-i *<IP address of remote controller>*
-u *<username>* **-p** *<password>*

Description

The **mxfencecfg** command is used in conjunction with **mxsetfence** to configure flexible fencing without running **mxconfig**. The **mxsetfence** command is run first, as described under “mxsetfence – configure fencing” on page 30.

If you are using the fcsan fence agent, run the **mxfencecfg** command with no parameters, as no additional configuration is needed.

For the webmgt fence agent, specify the appropriate fence parameters:

-r
Set the fencing action to RESET.

-s
Set the fencing action to SHUTDOWN.

-v <*vendor*>
Specify the vendor for the remote controller

-i <*IP address of remote controller*>
Specify the IP address of the remote controller.

-u <*username*>
Specify the username needed to access the remote controller.

-p <*password*>
Specify the password needed to access the remote controller.

mxinit – manage HP Clustered File System processes

Synopsis

```
/opt/hpcfs/bin/mxinit [-s, --start]
                      [-x, --stop]
                      [-H, --hard]
                      [-g, --status]
                      [-L, --load-mod]
                      [-U, --unload-mod]
                      [-f file, --alt-config-file file]
                      [-m, --monitor]
                      [-M, --no-monitor]
                      [--hba-status]
                      [--status]
                      [--version]
                      [-v, --verbose]
                      [-h, -?, --help]
```

Description

The **mxinit** utility is started automatically as a daemon on each server and monitors all HP Clustered File System processes running there. (You can start another instance of **mxinit** to perform other tasks provided by the utility.)

If a HP Clustered File System process dies unexpectedly on a server, **mxinit** will restart it if possible. However, certain process failures may force a restart of that particular server.

When you invoke **mxinit** to start HP Clustered File System, by default it continues running and monitors processes. If you do not want **mxinit** to monitor processes, invoke it with the **-M** (or **--no-monitor**) option. It will then exit after it completes the options you specified.

Typically, you should use the **pmxs** script to start or stop HP Clustered File System. However, if you want to see verbose output during the start or stop operation, you can run **mxinit** manually with the **--verbose** option.

mxinit performs its actions according to a set of default values. You can use the */etc/hpcfs/mxinit.conf* configuration file to override the default values. The file describes the available options and the required format. We recommend that you change this file only at the request of HP personnel.

The **mxinit** options are:

-s, **--start**
Start the HP Clustered File System processes.

-x, **--stop**
Gently stop the HP Clustered File System processes. **mxinit** first attempts to unmount PSFS filesystems. If the unmount fails, the gentle stop operation will also fail.

If you specify both **--stop** and **--hard**, the **mxinit** command first attempts the **--stop** operation. If it fails, **mxinit** then executes the **--hard** operation.

-H, **--hard**
Perform a hard, immediate stop of the HP Clustered File System processes. **mxinit** first attempts to terminate any applications accessing PSFS filesystems. It then unmounts the filesystems, terminates the HP Clustered File System processes, and unloads HP Clustered File System modules.

-L, **--load-mod**
Load all HP Clustered File System modules.

-U, **--unload-mod**
Unload all HP Clustered File System modules.

-f *file*, **--alt-config-file** *file*
Use the specified configuration file instead of the default configuration file (*/etc/hpcfs/mxinit.conf).*

-v, **--verbose**
Print verbose output about each step of the **mxinit** operation.

--version
Display the version of HP Clustered File System.

-m, **--monitor**
Explicitly tell **mxinit** to monitor processes. This is the default when **mxinit** is invoked to start HP Clustered File System.

-M, **--no-monitor**
Explicitly tell **mxinit** not to monitor processes.

--hba-status
Display the state of the FibreChannel host bus adapter drivers.

--status
Display the status of HP Clustered File System processes and modules. Following is an example.

```
$  status
HP Clustered File System Status:
Name                pid / status
mxinit              15930 mxinit: Monitoring MxS processes
mxlog               Loaded
grpcommd            15885
mxlogd              15887
```

```
pswebsvr            15890
clusterpulse        15892
panpulse            15912
psd_devfs           Loaded
psfs                Loaded
dlm                 15913
sanpulse            15917

FibreChannel adapter module status:
qla2300 - QLogic 2300 FibreChannel Adapter, is Loaded
```

The PID is displayed for running processes; “Stopped” is displayed for processes that are not running. For modules, the status specifies whether the module is loaded. The “FibreChannel adapter module status” section displays status for the FibreChannel adapter modules installed on the system.

mxlogd – log daemon

Synopsis

/opt/hpcfs/sbin/mxlogd

Description

The mxlogd daemon manages global error and event messages. It is used internally by HP Clustered File System and should never be run directly.

mxlogger – add a log message

Synopsis

mxlogger -e *<entity>* **-l** *<level>* [**-G**|**-L**] *<log-text>*

mxlogger [**-h**]

Description

You can use the **mxlogger** command to add your own messages to the *matrix.log* file. For example, you may want to add messages about the state of your applications.

The arguments are:

-h
Print a help message.

-e *entity*

The type of message that you are adding. *entity* can be USER1, USER2, USER3, USER4, USER5, USER6, USER7, or USER8. You will need to determine how you want to use these entities.

-l *level*

The severity of the message. *level* can be ERROR, WARN, INFO, EVENT, FATAL, STATE, TRACE, or DEBUG.

-G|L

-G specifies that the message to be added is global; **-L** specifies that it is local. The default is local.

log-text

The text of the message. If *log-text* contains special characters or spaces, it must be enclosed in quotation marks.

The following command adds a local message to the *matrix.log* file:

```
mxlogger -l info -e User2 "hello, world."
```

The message appears like this in the log file:

```
Server      Level  Date/time            Facility Entity Message
192.168.0.1 [Info] [2001-10-07 14:16:27] User    USER2  hello, world
```

mxmpconf – manage membership partitions

Synopsis

```
/opt/hpcfs/bin/mxmpconf
```

NOTE: HP Clustered File System cannot be running when you use **mxmpconf**. To stop the cluster, use the following command:

```
# /etc/init.d/pmxs stop
```

Description

The **mxmpconf** utility starts an ASCII interface that can be used to create a new set of membership partitions or to repair the existing partitions

Membership partitions control access to the SAN and store the device naming database, which includes the global device names for SAN disks imported into the cluster. Each server in the cluster has a membership partition file, which is called the "local MP list."

This file specifies the locations of the membership partitions. Each membership partition also has its own MP list. Because the membership partitions control access to the SAN, it is important that all servers in the cluster have the same view of where the partitions are located. The **mxmpconf** utility can be used to repair any problems if a failure causes servers to have inconsistent views of the membership partitions.

For detailed information about **mxmpconf**, see the *HP StorageWorks Clustered File System Administration Guide*.

Setup Option

The Setup option allows you to create a new set of membership partitions. You can select up to three SAN partitions to be used as membership partitions. It is recommended that the partitions be on separate disks.

Repair Menu

The Repair Menu lists the current membership partitions according to the MP file maintained on the server where you are running the utility.

Membership partitions are either active or inactive. The current membership partitions are active. There can also be old membership partitions in the cluster that are now inactive.

Membership Partition Status

The Repair Menu reports the status of each membership partition. The status will be one of the following:

OK. The membership partition is included in the local membership partition list. This is the normal status.

NOT FOUND. The **mxmpconf** utility cannot find the device containing the membership partition.

INACCESSIBLE. The **mxmpconf** utility cannot access the device containing the membership partition.

CORRUPT. The partition is not valid.

MISMATCH. The membership partition is valid but its MP list does not match the server's local MP list.

If the status is NOT FOUND or INACCESSIBLE, there may be a problem with the disk or with another SAN component. When the problem is repaired, the status should return to OK.

If the status is CORRUPT, you should resilver the partition. This step copies the membership data from a valid membership partition to the corrupted partition.

NOTE: The membership partition may have become corrupt because it was used by another application. Before resilvering, verify that it is okay to overwrite any existing data on the partition.

If the status is MISMATCH, you will need to determine which membership partitions are correct (either the partitions specified in the local MP list, or the partition labeled MISMATCH) and resilver accordingly.

Repair Menu Options

The Repair Menu includes the following options:

Resilver

Asks you to select a membership partition and then copies the data from that partition to the other membership partitions.

NOTE: If you resilver from a partition that is labeled MISMATCH, the operation may initialize partitions that are not currently membership partitions; any existing data on those partitions will be overwritten. Use the Display option to see the membership partition lists for the current membership partitions.

Add

Allows you to select a new membership partition. This operation configures the new partition as a membership partition, copies the data on the existing membership partitions to the new partition, and updates the local MP list and the lists on the existing membership partitions. This option appears only if there are fewer than three membership partitions. All membership partitions must have a status of OK.

Remove

Allows you to remove an existing membership partition. This operation removes the membership data and formatting from the partition. It also deletes the partition from the local MP list and from the membership partition lists on the remaining membership partitions.

Replace

Allows you to remove an existing membership partition and then to select a partition to replace it. (When you have completed your selections, choose "Done".) All membership partitions must currently have a status of OK.

Display

Shows the local membership partition list on the server where you are running **mxmpconf** and also displays the lists located on the disks containing the membership partitions. The output also includes the device database records for the disks containing the membership partitions.

Search

Searches the SAN for all partitions that appear to be membership partitions. The output includes each membership partition found by the search and specifies whether the partition is active or inactive. The output also displays the membership lists from the membership partitions and the database records for the partitions.

Inactivate

Inactivates membership partitions that are marked as active but are not part of the current set defined by the membership partition list. This option is useful if the cluster includes old membership partitions that are marked active or if you want to import a disk that contains an active membership partition.

Clear the Host Registry

This option removes all entries from the server registry.

CAUTION: Before clearing the server registry, be sure to reboot or power off any servers that were previously removed from the cluster and no longer had access to the SAN. After the servers have been rebooted, they can safely access the SAN. (If the servers are not rebooted, it is possible for them to corrupt filesystems.)

mxmpio – monitor or manage MPIO devices

Synopsis

/opt/hpcfs/bin/mxmpio *command* [*options...*] [*parameters ...*] [*PSD-devices ...*]

Description

HP Clustered File System uses multipath I/O (MPIO) to eliminate single points of failure. A cluster can include multiple FibreChannel switches, multiple FC ports per server, and multiported SAN disks. This configuration provides multiple paths for I/O between cluster servers and SAN disks.

When you start HP Clustered File System, it automatically configures all paths from each cluster server to the storage devices. On each server, it then uses the first path it discovered for I/O with the SAN devices. If that path fails, HP Clustered File System automatically fails over the I/O to another path.

The **mxmpio** command can be used to display status information about MPIO devices or to control the path used for I/O. With the exception of **enableall** and **disableall**, all commands take PSD device names to operate on specific devices. If you do not specify a PSD device, the command operates on all devices.

The options are:

-l

Long option. Provides more detail about targets. This option does not apply to **enableall** or **disableall**.

-u
Display I/O latencies in microseconds instead of the default milliseconds. This option applies only to the **iostat** command.

The commands are:

enable/disable
Enable or disable MPIO failover on the specified devices.

enableall/disableall
Globally enable or disable MPIO failover on this node.

timeout *value*
Set the timeout on the specified device.

active *target*
Set the active target on the specified device.

mpiostat
List the number of transient errors for each target and show the number of failovers and fatal errors for each device.

mpioload [*interval* [*count*]]
Shows the load for each target (SCSI command I/Os) and total for the PSD device (block layer I/Os), number of failovers, and fatal errors for each device.

iostat [*interval* [*count*]]
Show general I/O statistics for each device.

iostat [*interval* [*count*]]
Dump general I/O statistics for each device in a raw format.

Enable or Disable Failover for a Server

A cluster server can use multiple FC ports to connect to the FibreChannel fabric. If the FC port or FC switch currently being used for I/O should fail, HP Clustered File System will by default fail over the I/O to another FC port/FC switch path.

You can use the following command to control whether this failover behavior can occur on a particular node. Run the command on the server where you want to change the failover behavior.

```
# mxmpio enableall|disableall
```

HP Clustered File System starts with failover enabled.

Enable or Disable Failover for a PSD Device

When a failure occurs in the I/O path to a particular PSD device, HP Clustered File System will by default fail over to another I/O path. You can use the following command to control whether this failover behavior can occur for specific PSD devices. HP Clustered File System starts with failover enabled.

```
# mxmpio enable|disable [<PSD-device ...>]
```

Specify the Path for I/O

If you are troubleshooting problems with an I/O path, you may want to direct the I/O to another I/O path. You might also want to manually balance I/O paths on a per-server basis across the cluster. This is possible because **mxmpio**'s controls are server specific.

You can use the following command to specify either a particular HBA or a PSD device. HP Clustered File System will then fail over the I/O to the path that includes the specified device. In the command, *PSD-device* is specified by the base name of the device path, such as **psd2p1** (not **/dev/psd/psd2p1**).

```
# mxmpio active <target> <PSD-device>
```

target can be one of the following values:

I	A numerical index on the PSD device target array (0..).
M,m	A decimal major/minor number identifying the host adapter.
M:m	A hexadecimal major/minor number identifying the host adapter.

scsi*N*/*C*	A scsi bus ID. *N* is the scsi host adapter ID (0..) and *C* is the scsi host channel number (0..) as defined in */proc/scsi/scsi* and */proc/scsi/*/[0-9]*. If *C* is omitted, zero is assumed.
sd*N***s***P*	A device node in */dev*. The value is converted to a major/minor number identifying the corresponding host adapter.

With the exception of *I* (the array index), the value specified is converted to the corresponding host adapter/channel before being used to select the target.

An Example of Changing the I/O Path

In this example, we will change the target for a device. The **mxmpio status -l** command identifies the path currently being used by each device. That path is labeled "active." The following output shows that device psd2p1 is active on target 1.

```
# /opt/hpcfs/sbin/mxmpio status -l
MPIO Failover is globally enabled
            Failover  Timeout  Targets
    psd1    enabled    30000   0. (41:50) scsi2/0/2/19 (active)
                               1. (08:90) scsi1/0/2/19
    psd1p1  enabled    10000   0. (41:51) scsi2/0/2/19
                               1. (08:91) scsi1/0/2/19 (active)
    psd1p2  enabled    30000   0. (41:52) scsi2/0/2/19 (active)
                               1. (08:92) scsi1/0/2/19
    psd2    enabled    30000   0. (41:10) scsi2/0/1/20 (active)
                               1. (08:50) scsi1/0/1/20
    psd2p1  enabled    10000   0. (41:11) scsi2/0/1/20
                               1. (08:51) scsi1/0/1/20 (active)
    psd2p2  enabled    30000   0. (41:12) scsi2/0/1/20 (active)
                              1. (08:52) scsi1/0/1/20
```

Now use the **mxmpio** command to change the path for psd2p1 to target 0:

```
# /opt/hpcfs/sbin/mxmpio active 0 psd2p1
```

To verify the change, run the **mxmpio status -l** command again. In the following output, device psd2p1 is now active on target 0.

```
# /opt/hpcfs/sbin/mxmpio status -l
MPIO Failover is globally enabled
            Failover  Timeout  Targets
    psd1    enabled    30000   0. (41:50) scsi2/0/2/19 (active)
                               1. (08:90) scsi1/0/2/19
    psd1p1  enabled    10000   0. (41:51) scsi2/0/2/19
                               1. (08:91) scsi1/0/2/19 (active)
    psd1p2  enabled    30000   0. (41:52) scsi2/0/2/19 (active)
                               1. (08:92) scsi1/0/2/19
    psd2    enabled    30000   0. (41:10) scsi2/0/1/20 (active)
                               1. (08:50) scsi1/0/1/20
    psd2p1  enabled    10000   0. (41:11) scsi2/0/1/20 (active)
                               1. (08:51) scsi1/0/1/20
    psd2p2  enabled    30000   0. (41:12) scsi2/0/1/20 (active)
                              1. (08:52) scsi1/0/1/20
```

Display Status Information

The **status** command displays MPIO status information, including the timeout value, whether MPIO is enabled (globally and per-device), and any targets specified with the **active** command. Use the **-l** option to display more information about the targets, as in the above example.

```
$ mxmpio status
```

In contrast with the earlier example, this example shows output for a system that is not MPIO:

```
# /opt/hpcfs/sbin/mxmpio status -l
MPIO Failover is globally enabled
            Failover  Timeout  Targets
    psd1    enabled    30000   0. (41:50) scsi2/0/2/19
    psd1p1  enabled    10000   0. (41:51) scsi2/0/2/19
    psd1p2  enabled    30000   0. (41:52) scsi2/0/2/19
    psd2    enabled    30000   0. (41:10) scsi2/0/1/20
    psd2p1  enabled    10000   0. (41:11) scsi2/0/1/20
    psd2p2  enabled    30000   0. (41:12) scsi2/0/1/20
```

Set the Timeout Value

The default timeout period for PSD devices is 30 seconds. If you need to modify this value for a particular PSD device, use the following command. *value* is in milliseconds; however, the smallest unit is 10 milliseconds. A value of zero disables timeouts.

```
# mxmpio timeout value [PSD-device]
```

Show Number of Transient Errors

The **mpiostat** command lists the number of transient errors for each target and shows the number of failovers and fatal errors for each device. Values represent the number of respective events since the device was bound. If devices have not been rebound since boot, then this represents the number of events since boot.

Show Load Statistics

The **mpioload** command shows the load (number of pending I/Os, both total and raw) for each target (SCSI command I/Os) and total for the PSD device (block layer I/Os), number of failovers, and fatal errors for each device.

The syntax is:

mpioload [*interval* [*count*]]

interval is the number of seconds between samplings. The default is one second. *count* is the number of samples to make; the default is to sample indefinitely.

Values for failovers and fatal errors represent the number of respective events since the device was bound. Values for load are the instantaneous number of I/Os queued at the time of the sample. Raw I/Os are counted as requests (calls to read()/write()) while total (block layer) I/Os are per buffer_head structure (i.e., call to make_request()).

Display General I/O Statistics

The **iostat** command displays general I/O statistics for each device. Specific targets of PSD devices are not monitored individually. The syntax is:

iostat [*interval* [*count*]]

interval is the number of seconds between samplings. The default is one second. *count* is the number of samples to make; the default is to sample indefinitely.

The information displayed for each interval includes the number of I/Os queued (total block and raw), minimum and maximum latency, count of I/Os, and average latency. The statistics are organized by I/O, with only actively used sizes shown.

Latencies are in milliseconds. The minimum and maximum latency is reset every interval. The number of I/Os queued is an instantaneous value that is taken at the time of the sample.

NOTE: If *interval* is too long, the average latency computation may be affected by 32-bit computer wrap-around. A total of 71 minutes of I/O may be measured before counter wrap-around. This does not necessarily equate to 71 minutes in realtime.

Dump I/O Statistics in Raw Format

The **rawstat** command dumps general I/O statistics for each device in raw format. The output is typically passed to programs or shell scripts for further processing. Numbers are the raw counter values, not computed per sample. The syntax for the command is:

rawstat [*interval* [*count*]]

interval is the number of seconds between samplings. The default is one second. *count* is the number of samples to make; the default is to sample indefinitely.

The output consists of ASCII text delimited by carets. The fields are:

1. PSD device name
2. Sample number
3. Sampling interval
4. Number of targets
5. Current active target
6. Number of I/O sizes
7. First I/O size (power of 2)
8. Number of I/Os queued (includes raw)
9. Number of raw I/Os queued
10. Number of MP failovers
11. Number of MP fatal errors

12... Per-target I/O statistics in tuples, or groups of two numbers. (The number of targets is indicated in field 4.) Each tuple consists of the following fields for each target:

- SCSI I/Os queued
- Transient failures

Note that the "SCSI I/Os queued" numbers are for the underlying disk, not the partition. PSD devices that share the same underlying disk will share the same numbers here.

12+$4*2... I/O statistics in quads, or groups of four numbers. (The number of quads appears in field 6, "Number of I/O sizes"). Each quad consists of the following fields for each I/O size:

- Minimum latency
- Maximum latency
- Count of I/Os
- Total latency for all I/Os

The minimum and maximum latency are reset every interval. Latencies are in microseconds.

mxnlmconfig – enable or disable

Synopsis

/opt/hpcfs/mxnlmconfig -q|-e|-d

Description

NLM is the locking protocol used by NFS. By default, it is disabled in CFS-Linux. If necessary, NLM can be enabled; however, you should be aware of the following caveat:

- File locks granted by the NFS server are cluster-coherent. When a failover occurs, the locks are released by the original server and the client automatically reclaims them on the new server (the backup node). However, during the period after the lock is released, another client or application may compete for and win the lock.

Some NFS clients will return an error to the client applications if the lock cannot be reclaimed. Other clients (for example, the Linux 2.6 NFS client) will not return any error. If no error is returned by the client, the application may proceed under the false assumption that the lock has been granted. Data corruption may be the result.

To prevent this situation, locking should be enabled only if your clients are partitioned so that all clients needing a particular lock are using the same Virtual NFS Service IP address. If a failover occurs, all of the clients will lose their locks. They can then reclaim those same locks on the new node without conflicts from outside clients.

The options are as follows:

-q
Show the current status of NLM locking in the cluster (either enabled or disabled).

-e
Enable NLM locking in the cluster. No reboot is necessary; the change is effective almost immediately and may affect clients.

-d
Disable NLM locking in the cluster. No reboot is necessary; the change is effective almost immediately and may affect clients.

mxpasswd – add users or change passwords

Synopsis

/opt/hpcfs/bin/mxpasswd

Description

To add a new HP Clustered File System user and password, use the following syntax, where *user* and *password* are enclosed in curly braces. You must be user *root*.

```
# mxpasswd
mxpasswd> {user}{password}
```

To change an existing password, use the following syntax. Users can change their own passwords. If you are *root*, you can change any user password without specifying the old password.

```
$ mxpasswd
mxpasswd> {user} {new_password} {old_password}
```

mxperftool – view performance counters

Synopsis

/opt/hpcfs/tools/mxperftool

Description

This command should be run only at the request of HP personnel.

mxregd – CFS-Linux daemon

Synopsis

/opt/hpcfs/sbin/mxregd

Description

The mxregd daemon manages the CFS-Linux configuration. It is used internally and should never be run directly.

mxregtool – explore the mxreg datastore

Synopsis

/opt/hpcfs/tools/mxregtool

Description

This command is used internally and should not be run directly.

mxsancheck – check server's SAN access

Synopsis

/opt/hpcfs/bin/mxsancheck

Description

The **mxsancheck** command determines whether a server has SAN access and is ready to mount filesystems. The command is intended to be used in scripts and returns 0 on success and 1 on failure.

mxsanconf – configure FC switches

Synopsis

/opt/hpcfs/sbin/mxsanconf *<FC-switch ...>*

Description

When a cluster is configured to use fabric-based fencing, **mxconfig** runs the **mxsanconf** command on each node to configure the list of FibreChannel switches that will be managed by HP Clustered File System. The command creates or updates the files */etc/hpcfs/psSAN.cfg* and */var/hpcfs/FCswitches*.

<*FC-switch*> is either the name or IP address of a switch to be managed. All switches to be configured must be specified in the same command. The node must be unfenced when the command is run. If the <*FC-switch*> parameter is not specified, the command uses the FC switches listed in */var/hpcfs/FCswitches* (if that file exists).

In general, this command should not be run directly.

mxsanlk – show SAN ownership locks

Synopsis

/opt/hpcfs/tools/mxsanlk

Description

HP Clustered File System uses a set of disk-based data structures called SANlocks to protect filesystem integrity. If a problem causes a cluster to split into two or more network partitions, the SANlocks ensure that only one of the resulting network partitions has access to the SAN. Each SANlock is stored in a membership partition. Before a cluster can begin accessing the SAN, it must first acquire a majority of the SANlocks. The SANlocks are acquired in order.

mxsanlk displays the status of the SANlock stored in each membership partition. It can be used to determine whether any of the membership partitions need to be repaired. Also, if a network partition occurs, **mxsanlk** can be used to determine which network partition has control of the SAN.

Following is some sample output. The command was issued on host 10.10.30.3. The SDMP administrator is the administrator for the cluster to which the host belongs. There are three membership partitions.

```
# mxsanlk
This host:                       10.10.30.3
This host's SDMP administrator: 10.10.30.1

Membership Partition    SANlock State
--------------------    -------------
/dev/rpsd/psd1p1         held by SDMP administrator
/dev/rpsd/psd2p1         held by SDMP administrator
/dev/rpsd/psd3p3         held by SDMP administrator
```

Any of these messages can appear in the "SANlock State" column.

held by SDMP administrator
: The SANlock was most recently held by the SDMP administrator of the cluster to which the host where **mxsanlk** was run belongs.

trying to lock, last held by host *X.X.X.X*
: The SANlock was most recently held by host *X.X.X.X* and may still be held by that host. The host on which **mxsanlk** was run is trying to acquire the SANlock.

cannot access
: The host on which **mxsanlk** was run is unable to access the SANlock. The membership partition may need to be repaired.

trying to lock, not yet committed by owner
: The SANlock is either not held or has not yet been committed by its holder. The host on which **mxsanlk** was run is trying to acquire the SANlock.

unlocked, trying to lock
: The SANlock does not appear to be held. The host on which **mxsanlk** was run is trying to acquire the SANlock.

unlocked
: The SANlock does not appear to be held. If a host holds the SANlock, it has not yet committed its hold.

initiating sdmp, not yet examined
: This is a transitional state. It indicates that the sdmp process responsible for the SANlock has been started but has not yet accessed the SANlock.

sdmp process hung
: The SDMP process responsible for the SANlock is unresponsive.

trying to lock, sdmp process hung
: The host on which **mxsanlk** was run is trying to acquire the SANlock but the SDMP process responsible for the SANlock is unresponsive.

locked, sdmp process hung
: The host on which **mxsanlk** was run held the SANlock but the SDMP process responsible for the SANlock is now unresponsive.

lock is corrupt, will repair
: This transitional state occurs after the SDMP has detected that the SANlock has been corrupted but before it has repaired the SANlock.

trying to lock (lock is corrupt, will repair)
: The host on which **mxsanlk** was run is trying to acquire the SANlock. The SANlock was corrupted but will be repaired.

locked (lock is corrupt, will repair)
: The host on which **mxsanlk** was run holds the lock. The SANlock was corrupted but will be repaired.

If a membership partition cannot be accessed, use the **mxmpconf** program to correct the problem.

When you invoke **mxsanlk**, it checks for the Storage Device Monitor Pulse (SDMP) daemon. This daemon is responsible for grabbing and maintaining the locks on the membership partitions. Depending on the status of the SDMP daemon, you may see one of the following messages:

```
Checking for SDMP activity, please wait...
Still trying...
The SDSMP is inactive at this host.
The SDMP appears to be inactive at this host.
```

If the SDMP daemon is not responding on the host, wait a few seconds and retry the command. If the command continues to fail, shut down the cluster and then restart it. This step should restart the SDMP daemon. If you continue to have problems, contact HP Technical Support.

mxsetfence – configure fencing

Synopsis

/opt/hpcfs/bin/mxsetfence *<fence_agent>*

Description

The **mxsetfence** command is used in conjunction with the **mxfencecfg** command to configure flexible fencing without running **mxconfig**.

The **mxsetfence** command is run first, and initializes the fence configuration file */etc/hpcfs/fence.conf*. It also creates the symbolic link **/opt/hpcfs/bin/mxfencecfg**, which is linked to the fence configuration utility for the specified fence agent.

The fence agents are:

fcsan
Fabric fencing.

webmgmt
Server-based fencing.

After running **mxsetfence**, you will need to configure the fence agent with **mxfencecfg**. See "mxfencecfg – configure the fence agent" on page 8 for more information.

mxsetsecret – set the network secret password

Synopsis

/opt/hpcfs/bin/mxsetsecret -f [**--filename**] <*filename*> <*secret*>

Description

This command is typically run by **mxconfig** and should be run manually only at the request of HP personnel.

panpulse – PanPulse daemon

Synopsis

/opt/hpcfs/sbin/panpulse

Description

The PanPulse daemon selects and monitors the network to be used for the administrative network, verifies that all hosts in the cluster can communicate with each other, and detects any communications problems. This daemon is used internally by HP Clustered File System and should never be run directly.

pmxs – start or stop HP Clustered File System or view status

Synopsis

/etc/init.d/pmxs start
/etc/init.d/pmxs stop
/etc/init.d/pmxs restart
/etc/init.d/pmxs status

Description

HP Clustered File System runs on each server in the cluster. When a server is booted to run-levels 3 or 5, HP Clustered File System is started automatically by the script **/etc/init.d/pmxs**. If you need to start, stop, or restart HP Clustered File System on a particular server, invoke the **pmxs** script with the appropriate argument.

The **pmxs** script calls the **mxinit** utility.

For a start operation, **mxinit** starts the HP Clustered File System processes in the correct order and loads the kernel modules. For a stop operation, it stops the HP Clustered File System processes, unloads the kernel modules, and performs cleanup tasks.

The **status** option displays the same information as the **mxinit --status** command. You do not need to be user *root* to run the command.

PSANcfg – manage ports, community string

Synopsis

/opt/hpcfs/sbin/PSANcfg [**-hu**] [-[**lL**] *<Port WWN>*] [*switch ...*]

Description

The **PSANcfg** command can be used to add or remove local HBA port information in the HP Clustered File System configuration and to unfence ports on FibreChannel switches. The options are:

-l, **-L**
: The **-l** command adds the specified HBA port to the list of local ports; the **-L** command removes the specified port. The **mxsanconf** command invokes **PSANcfg** with these options; they should not be run directly.

-u *switch ...*
: Unfence all local ports on the specified FC switches.

-h
: Print a usage message.

PSANinfo – show FC logins, Naming Database

Synopsis

/opt/hpcfs/sbin/PSANinfo [**-n**] [[**-s**]|[**-u**]] *<switchname>*

Description

The **PSANinfo** command can be used to print the Naming Database or to get a snapshot of logins for a particular FibreChannel switch.

The options are:

-n
Print the contents of the Naming Database.

-s
Wait <*s*> seconds between probes of the FC switch.

-u
Wait <*u*> microseconds between probes of the FC switch.

If no options are specified, **PSANinfo** displays the status of the switch one time only.

Following is a sample snapshot:

```
Switch 10.10.11.240 : 1588 Fibre Channel Switch.
   IP addr 10.10.11.240 WWN 10000060693025CD Fabric ID 10000060693025CD
   1 module
   Module 1 : 10000060693025CD state 1 has 8 ports
      Port  0 : oper 2 admin 1
      Port  1 : oper 1 admin 1 Fabric ID 0x111100 attached to 210100E08B255640
      Port  2 : oper 1 admin 1 Fabric ID 0x111200 attached to 210000E08B026C65
      Port  3 : oper 2 admin 1
      Port  4 : oper 1 admin 1 Fabric ID 0x111400 attached to 210000E08B056F21
      Port  5 : oper 1 admin 1 Fabric ID 0x111500 attached to 200B00A0B80F2851
      Port  6 : oper 2 admin 1
      Port  7 : oper 2 admin 1
Poll time: 0.37
```

psdctl – manage device bindings

Synopsis

/opt/hpcfs/tools/psdctl

Description

This command should be run only at the request of HP personnel.

psfsck – check and repair a PSFS filesystem

Synopsis

/opt/hpcfs/sbin/psfsck [**-aixoV**] [**-l** *<logfilename>*] [**--check**] [**--rebuild-tree**] [**--rebuild-sb**] [**--interactive**] [**--quiet**] [**--logfile** *<filename>*] [**--fix-fixable**] [**--fix-non-critical**] *<device>*

Description

The **psfsck** utility looks for a PSFS filesystem on a device, replays transactions that are to be replayed, and either checks or repairs the filesystem.

device is a psd or psv device and is specified as follows:

- For a psd device partition, the device is specified as **/dev/psd/psd***XXX***p***YY*, where XXX is the drive number and YY is the partition number. For example, **/dev/psd/psd6p4** specifies partition 4 on disk psd6.
- For a non-partitioned psd device, the device is specified as **/dev/psd/psd***XXX*, where *XXX* is the drive number. For example, /**dev/psd/psd5**.
- For a psv device, the device is specified as **/dev/psv/psv***XXX*, where *XXX* is the volume number. For example, **/dev/psv/psv1**.

When **psfsck** is running in check mode (the default action), it will attempt to fix any corruptions that can be repaired without **--rebuild-tree**. The types of corruption that can be fixed include: bad pointers to data blocks, incorrect st_size and st_blocks in a directory, directory entries pointing to nowhere, incorrect file sizes and modes, and objectid sharing.

NOTE: The **psfsck** utility requires exclusive access to the device. If it cannot obtain exclusive access, it will exit with an "ebusy" message.

The options are:

--check
Check filesystem consistency. This is the default action.

--rebuild-tree
Rebuild the filesystem tree using leaf nodes found on the device. Normally you only need this option if check mode (without the **-no-modify** flag) reports errors that can be fixed only by **--rebuild-tree**. You are strongly encouraged to make a backup copy of the whole partition before attempting to rebuild.

If **--rebuild-tree** encounters any files that had been open and unlinked, **psfsck** will move the files into the *lost+found* directory.

--rebuild-sb
Rebuild the superblock.

--interactive, **-i**
Stop **psfsck** after each pass is completed.

--quiet, **-q**
Prevent **psfsck** from reflecting its progress.

--logfile *filename*, **-l** *logfilename*
Tell **psfsck** to place information about any corruption it finds into the specified logfile instead of sending it to *stderr*.

--no-modify, **-n**
Check the filesystem in read-only mode. Prevents **psfsck** from replaying the journal and/or fixing any corruption. If errors are found, it is strongly recommended that you run **psfsck** again in check mode, without the **--no-modify** option, before running with the **--rebuild-tree** option.

The **-no-modify** option cannot be specified in addition to **--rebuild-tree** or **--rebuild-sb**.

-a
Cause **psfsck** to assume that it was called by **psfsck -A** and to return, even if the filesystem does not seem to be unmounted cleanly.

-y
Cause **psfsck** to answer "yes" to all questions.

psfsdebug – get PSFS filesystem image

Synopsis

/opt/hpcfs/tools/psfsdebug -p *<device>*
psfsdebug -P *<device>*

Description

When used to obtain a PSFS filesystem image, the **psfsdebug** command extracts all metadata from the specified filesystem and sends it to stdout. The **-p** option compresses the data, while the **-P** option does not.

<device> is either the special file corresponding to the psd or psv device or the corresponding local device if psd or psv drivers are not loaded.

-p is the best option for a filesystem that is not corrupted, as the resulting output is smaller. With corrupted filesystems, there is a risk that the **-p** compression algorithm will be affected by the corruption and the unpacked metadata will not be identical to the original.

See "psfsunpack – unpack a PSFS filesystem image" on page 39 for a typical pack/unpack sequence.

psfslabel – label a PSFS filesystem

Synopsis

/opt/hpcfs/tools/psfslabel *<device> "<label>"*

Description

The **psfslabel** command adds a label to the specified device.

psfsresume – resume a suspended PSFS filesystem

Synopsis

/opt/hpcfs/tools/psfsresume *<directory_mountpoint>*

Description

The **psfssuspend** and **psfsresume** utilities affect the specified filesystem on all servers where it is mounted; however, the utilities should be executed on only one server in the cluster.

When you have completed your work with the suspended filesystem, use the **psfsresume** utility to resume the filesystem. Issue the **psfsresume** command from the server where you executed **psfssuspend**. You must be user *root*.

NOTE: If an attempt to mount the copied filesystem fails with an "FSID conflict" error, run the following command as user *root*. In the command, <*device*> is the partition, such as **/dev/psd/psd1p7**, that contains the copied filesystem, and <*label*> is the name that should be used to identify the filesystem.

/opt/hpcfs/tools/psfslabel *<device> "<label>"*

psfssema – manage cluster-wide file locks

Synopsis

/opt/hpcfs/bin/psfssema -i *<filename>*
/opt/hpcfs/bin/psfssema -g *<filename>*
/opt/hpcfs/bin/psfssema -r *<filename>*
/opt/hpcfs/bin/psfssema -d *<filename>*

Description

The **psfssema** semaphore utility provides a simple synchronization mechanism for managing cluster-wide file locks. This utility can be used in shell scripts on different nodes of a cluster and takes advantage of the PSFS filesystem and its internode communication abilities. For example, you might want to use cluster-wide file locking in a Start or Stop script for a service or device monitor.

The options are:

-i *<filename>*
Create the PSFS command-line semaphore file <*filename*> and the supplemental file <*filename.pid*> if either file does not already exist. It then initializes the files to create the PSFS command-line semaphore. <*filename*> must be a path on a PSFS filesystem.

-g *<filename>*
Lock the PSFS command-line semaphore associated with <*filename*>, which is the name of a semaphore file created by **psfssema -i**.

If the semaphore is already locked by another node in the cluster, the calling process blocks until the semaphore becomes available.

An error is returned if <*filename*> does not exist or has not been initialized by **psfssema -i**, or if <*filename.pid*> does not exist.

-r *<filename>*
Unlock the PSFS command-line semaphore associated with <*filename*>, which is the name of a semaphore file created by **psfssema -i**. If other nodes are blocked on the semaphore when **psfssema-r** is called, one of the blocked **psfssema -g** processes will return successfully.

-d *<filename>*
Delete the PSFS command-line semaphore associated with <*filename*>, which is the name of a semaphore file created by **psfssema -i.** It also deletes <*filename*> and <*filename.pid*>.

psfssuspend – suspend a PSFS filesystem

Synopsis

/opt/hpcfs/tools/psfssuspend *<directory_mountpoint>*

Description

The **psfssuspend** utility suspends a PSFS filesystem in a stable, coherent, and unchanging state. While the filesystem is in this state, you can copy it for backup and/or archival purposes. When copying directly from a suspended device, be sure to use the raw device (*/dev/rpsd/...*) to ensure that all blocks copied are up-to-date.

The filesystem is essentially unusable while it is suspended; however, applications that can tolerate extended waits for I/O do not need to be terminated.

The **psfsresume** utility restores a suspended filesystem. The **psfssuspend** and **psfsresume** utilities affect the specified filesystem on all servers where it is mounted; however, the utilities should be executed on only one server in the cluster.

To suspend a filesystem, issue the following command on one server that has mounted the filesystem. You must be user *root*.

/opt/hpcfs/tools/psfssuspend *directory_mountpoint*

psfsunpack – unpack a PSFS filesystem image

Synopsis

/opt/hpcfs/tools/psfsunpack *<device>*

Description

The **psfsunpack** command unpacks a PSFS filesystem that was packed with the **psfsdebug** command. It reads the PSFS metadata from stdin and creates a corresponding filesystem on the given device.

Following is a typical execution sequence:

```
psfsdebug -p /dev/psd/psd1p2 > psd1p2.pack
<copy psd1p2.pack to an analysis machine>
psfsunpack /dev/sdb1 < psd1p2.pack
```

psvctl – manage dynamic volumes

Synopsis

/opt/hpcfs/tools/psvctl

Description

This command should be run only at the request of HP personnel.

pswebsrv – web server daemon

Synopsis

/opt/hpcfs/sbin/pswebsrv

Description

pswebsrv is the embedded web server daemon used by the Management Console and the **mx** utility.

This daemon is used internally by HP Clustered File System and should never be run directly.

resizepsfs – resize a PSFS filesystem

Synopsis

/opt/hpcfs/sbin/resizepsfs -s [+|-]*size*[**K**|**M**|**G**|**T**]

Description

The **resizepsfs** program can be used to increase the size of a PSFS filesystem. **resizepsfs** can grow a filesystem online if the filesystem is mounted on the server where the utility is invoked. Otherwise, the filesystem must be unmounted on all servers before it is resized.

The utility can be used with either **psd** or **psv** devices. Use this syntax to specify the device:

- For a psd device partition, the device is specified as **/dev/psd/psd***XXX***p***YY*, where XXX is the drive number and YY is the partition number. For example, **/dev/psd/psd6p4** specifies partition 4 on disk psd6.
- For a non-partitioned psd device, the device is specified as **/dev/psd/psd***XXX*, where *XXX* is the drive number. For example, /**dev/psd/psd5**.
- For a psv device, the device is specified as **/dev/psv/psv***XXX*, where *XXX* is the volume number. For example, **/dev/psv/psv1**.

This program does not change the size of the partition containing the filesystem. Instead, you will need to use a utility specific to your RAID subsystem to modify the size of the partition. You will need to deport the disk containing the filesystem before you modify the partitions.

CAUTION: Be sure to back up your data before using this program.

You can use the **-s** option to specify the new size for the filesystem. If you do not specify the size, the filesystem will grow to the size of the partition. The **-s** option can be used as follows:

- Specify the size in kilobytes, megabytes, gigabytes, or terabytes:

  -S *size*[**K**|**M**|**G**|**T**]

- Specify the amount (in kilobytes, megabytes, gigabytes, or terabytes) by which the filesystem should be increased:

  -s [**+**|**-**]*size*[**K**|**M**|**G**|**T**]

The following example increases the size of the filesystem by 1 GB.

```
resizepsfs -s +1G /dev/psd/psd6p4
```

sanconfigure – obsolete

This command is obsolete and will be removed in a future release.

sandiskinfo – show SAN disk information

Synopsis

/opt/hpcfs/bin/sandiskinfo [**-i**|**-u**|**-v**] [**-al**] [**-f**] [**-U**] [**--subdevices**] [**--dynvolumes**] [**--dynvol_properties** [*volname*]]

Description

The **sandiskinfo** command can display information for both imported and unimported SAN disks and also for dynamic volumes. Under normal operations, the **sandiskinfo** output should be the same on all servers in the cluster.

Disk Information

With no options, **sandiskinfo** displays the UID, vendor, model, and capacity of each imported disk and specifies the FC switch used to access the disk.

```
# sandiskinfo
Disk: /dev/psd/psd5
Uid: 20:00:00:04:cf:13:32:d1::0 SAN info: switch fcswitch port 7
Vendor:      SEAGATE ST336704FC Capacity: 34732M
```

Following are the commonly used options for imported and unimported disks:

sandiskinfo [**-i**|**-u**|**-v**] [**-al**] [**-f**] [**-U**]

The default is **-i**, which produces the output shown above for imported disks. The **-u** option produces the same output for unimported disks.

The **-U** option displays output in the format used by the Management Console. This option is used internally by HP Clustered File System and does not produce human-readable output.

Show Partition Information

The **-a** option also lists the partitions on each disk. When combined with **-u**, it displays partition information for unimported disks.

```
# sandiskinfo -a
Disk: /dev/psd/psd5
Uid: 20:00:00:04:cf:13:32:d1::0 SAN info: switch fcswitch port 7
Vendor:      SEAGATE ST336704FC Capacity: 34732M
 partition 01: size 3905M type Linux (83)
 partition 02: size  813M type Linux (83) (PSFS Filesystem)
 partition 03: size 7813M type Linux (83) (Membership Partition)
```

Show Local Device Information

The **-l** option displays the local device name for each disk, as well as the default disk information. When combined with **-u**, it displays local device names for unimported disks.

```
# sandiskinfo -al
Disk: /dev/psd/psd5
Uid: 20:00:00:04:cf:13:32:d1::0 SAN info: switch fcswitch port 7
Vendor:      SEAGATE ST336704FC Capacity: 34732M
Local Device Path:  /dev/sda
 partition 01: size 3905M type Linux (83)
 partition 02: size  813M type Linux (83) (PSFS Filesystem)
 partition 03: size 7813M type Linux (83) (Membership Partition)
```

Show Filesystem Information

The **-f** option displays existing PSFS filesystems on imported disks.

```
# sandiskinfo -f
Volume: /dev/psv/psv1            Size:   2439M  (PSFS Filesystem)
    Stripesize=0K
    Local Mount Point=/mnt
Volume: /dev/psd/psd1p6          Size:    490M  (PSFS Filesystem)
    Disk=20:00:00:04:cf:13:38:18::0 partition=06 type=Linux (83)
    Local Mount Point=(not mounted)
```

Show Available Volumes

The **-v** option lists available volumes on imported disks. These volumes are not currently in use for a PSFS filesystem or a membership partition.

```
# sandiskinfo -v
Volume: /dev/psd/psd5p1              Size:  3905M
Disk=20:00:00:04:cf:13:32:d1::0 partition=01 type=Linux (83)
Volume: /dev/psd/psd5p2              Size:  7386M
Disk=20:00:00:04:cf:13:32:d1::0 partition=01 type=Linux (83)
```

Options for Dynamic Volumes

The following **sandiskinfo** options apply only to dynamic volumes.

Show Available Subdevices

The **--subdevices** option lists subdevices that are available for use in constructing a dynamic volume.

```
# sandiskinfo --subdevices
Subdevice: 20:00:00:04:cf:13:38:18::0/2     Size:   1950M   psd1p2
Subdevice: 20:00:00:04:cf:13:38:18::0/7     Size:    490M   psd1p7
Subdevice: 20:00:00:04:cf:13:38:18::0/8     Size:    490M   psd1p8
Subdevice: 20:00:00:04:cf:13:38:18::0/9     Size:    490M   psd1p9
Subdevice: 20:00:00:04:cf:13:38:18::0/10    Size:    490M   psd1p10
Subdevice: 20:00:00:04:cf:13:38:18::0/11    Size:    490M   psd1p11
Subdevice: 20:00:00:04:cf:13:38:18::0/12    Size:    490M   psd1p12
Subdevice: 20:00:00:04:cf:13:38:18::0/13    Size:    490M   psd1p13
Subdevice: 20:00:00:04:cf:13:38:18::0/14    Size:    490M   psd1p14
```

Show Dynamic Volumes

The **--dynvolumes** option lists all dynamic volumes.

```
# sandiskinfo --dynvolumes
Dynamic Volume: psv1             Size:   2439M  Stripe=Unstriped
Dynamic Volume: psv2             Size:    490M  Stripe=32K
Dynamic Volume: psv3             Size:    490M  Stripe=8K
```

Show Properties for Dynamic Volumes

The **--dynvol_properties** [*volname*] option lists detailed properties for the specified dynamic volumes. *volname* is the **psv** name, such as *psv2*. If this option is omitted, the properties for all dynamic volumes are displayed.

```
# sandiskinfo --dynvol_properties
Dynamic Volume: psv1             Size:   2439M  Stripe=Unstriped
   Subdevice: 20:00:00:04:cf:13:38:18::0/5    Size:   490M  psd1p5
   Subdevice: 20:00:00:04:cf:13:38:18::0/2    Size:  1950M  psd1p2
Dynamic Volume: psv2             Size:    490M  Stripe=32K/optimal
   Subdevice: 20:00:00:04:cf:13:38:18::0/7    Size:  490M   psd1p7
Dynamic Volume: psv3             Size:    490M  Stripe=8K/optimal
   Subdevice: 20:00:00:04:cf:13:38:18::0/10   Size:  490M   psd1p10
```

sanlibmig.sh – migrate the SAN identity

Synopsis

/opt/hpcfs/tools/sanlibmig.sh

Description

The **sanlibmig.sh** script executes as part of the HP Clustered File System installation and should not be run manually.

sanpulse – SanPulse daemon

Synopsis

/opt/hpcfs/sbin/sanpulse

Description

The SanPulse daemon provides the cluster infrastructure for management of the SAN. It coordinates filesystem mounts, unmounts, and crash recovery operations. This daemon is used internally by HP Clustered File System and should never be run directly.

smds – show UI status

Synopsis

/opt/hpcfs/tools/smds

Description

This command should be run only at the request of HP personnel.

snapctl – manage snapshot operations

Synopsis

/opt/hpcfs/sbin/snapctl

Description

This command is used internally by HP Clustered File System and should not be run directly.

spctl – dump the SanPulse trace buffer

Synopsis

/opt/hpcfs/tools/spctl -l

Description

This command should be run only at the request of HP personnel.

spdebug – obtain SanPulse debug information

Synopsis

/opt/hpcfs/tools/spdebug

Description

This command should be run only at the request of HP personnel.

spstat – show cluster state information

Synopsis

/opt/hpcfs/tools/spstat

Description

This command should be run only at the request of HP personnel.

wmtest – test server-based fencing

Synopsis

/opt/hpcfs/tools/wmtest

Description

This command should be run only at the request of HP personnel.

2

mx Commands

The **mx** utility provides a command-line interface for administering a cluster and monitoring its operation.

Using the mx utility

The matrixrc file

The **mx** utility can be used both interactively and in scripts. Because cluster administration requires a password, HP Clustered File System uses an external configuration file named *matrixrc* to provide authentication. This file is required for **mx** operation and contains password information.

If you want to use the **mx** utility, create a *matrixrc* file that specifies your user names and passwords for each server that you want to access. The default location for the file is *$HOME/.matrixrc*. For security reasons, the file must not allow any permissions for group or other.

The entries in the *matrixrc* file have this format:

```
machine   user   password   default
```

- The first field, *machine*, is either the name or the IP address of the server.
- The second field, *user*, is the name of a user on that server, either *admin* for the administrator or another name for a read-only user.
- The third field is the *password*.

- The fourth field, **default**, specifies that this server will be connected to by default if a server name is not specified on the command line. One entry in the file must contain this field.

Blank lines and lines beginning with a # character are ignored.

For example, you could create a *matrixrc* file containing these entries:

```
acme1 admin    secret1
acme2 admin    secret2    default
```

When you issue the **mx** command without specifying a server name, as in the following example, it connects to the default machine, *acme2,* as user *admin* using the password *secret2*.

```
mx server status
```

To connect to a different server, you will need to specify its name on the command line. For example, the following command connects to the server *acme1* as user *admin* using the password *secret1*.

```
mx --matrix acme1 server status
```

By default, the **mx** utility uses the *matrixrc* file. However, you can specify a different configuration file with the **--config** option.

mx syntax

The **mx** utility has the following syntax:

```
mx [mx_options] class command [command_options]
```

The *mx_options* affect an entire **mx** command session. The options are as follows:

--help
Displays a command summary.

--matrix <*cluster*>
Specifies the cluster that you want to connect with. *cluster* can be any node in the cluster.

--config <*file*>
Specifies the configuration file to be consulted for server, user, and password information. The file must have the same format as *matrixrc.*

--file <*file*>
Executes the commands from the specified *file*. If you specify – instead of a *file*, the commands will be read from standard input. You can also specify **--continue** to tell **mx** to continue processing the batch file if an error occurs.

--prompt <*prompt*>
Specifies the prompt string that will be printed when **mx** is ready for another command. This option is useful when you initiate an interactive session with the option **--file -**. For example, if you invoke an interactive session with the following command, **mx** will print a % prompt when it completes a command.

```
mx --prompt '%' --file -
```

--numeric
Causes hosts to be specified by their numeric IP addresses instead of by their hostnames.

--user <*username*>
Specifies the user to be logged in.

Class syntax

The **mx** utility can configure and monitor the following classes of cluster objects.

Class	Cluster Object
device	Device monitor
disk	SAN disk
dynvolume	Dynamic volume
exportgroup	Export Group
fs	PSFS filesystem
matrix	The entire cluster
netif	Network interface
notifier	Notifier
server	Server
service	Service monitor

Class	Cluster Object
snapshot	Snapshot
vhost	Virtual host
vnfs	Virtual NFS Service

To specify a command affecting a class, use this syntax:

```
<class> <command> <arguments>
```

For example, the following command displays the status of servers that are currently up:

```
mx server status --up
```

You can specify **--help** to see a short command synopsis for each class.

mx device – device monitor commands

Use the following commands to configure device monitors or to display their status.

Add a device monitor:

```
mx device add --servers <server1>,<server2>,...
[optional-arguments] <devicename> ...
```

--servers specifies the server or servers that use the monitored device. The <*devicename*> can include up to 32 characters.

The arguments are:

`[--timeout <seconds>]`
The maximum amount of time to wait for a probe of the device to complete. For DISK, GATEWAY, and SHARED_FILESYSTEM device monitors, the default is five seconds. For CUSTOM device monitors, the default is 60 seconds.

`[--frequency <seconds>]`
The interval at which the monitor probes the device. For DISK and SHARED_FILESYSTEM device monitors, the default is 30 seconds. For GATEWAY device monitors, the default is five seconds. For CUSTOM device monitors, the default is 60 seconds.

[**--type CUSTOM**|**DISK**|**GATEWAY**|**SHARED_FILESYSTEM**]
The type of device monitor.

[**--probeSeverity nofailover**|**autorecover**|**noautorecover**]
The failover behavior for the monitor. **nofailover** prevents failover of virtual hosts when the monitored device fails. **autorecover** fails over the virtual hosts, and when the device is restored, fails the virtual hosts back to the original network interfaces. **noautorecover** fails over the virtual hosts but does not fail them back after the device is restored. The default is **autorecover**.

[**--parameters** <*parameters*>]
The available parameters depend on the type of monitor:

- DISK device monitor. The parameter is a partition on the disk. The monitor will attempt to read the first block on this partition to determine whether the disk is operating normally.
- GATEWAY device monitor. The IP address of the gateway device (such as a router). The IP address must be on a different subnet than the servers in the cluster.
- CUSTOM device monitor. The parameter is a probe script. The maximum length of the pathname is 512 characters.
- SHARED_FILESYSTEM device monitor. The first parameter is the volume (for example, psd1p6) containing the filesystem to be monitored. The second parameter is the name of a file that the monitor probe should open and attempt to read to determine the health of the filesystem. The filename should be relative to the mount point of the filesystem. When the filesystem is mounted, the mountpath will be prepended to the filename to determine the complete filename path that should be probed. The second parameter is optional.

[**--ordering serial**|**parallel**]
Whether HP Clustered File System enforces a strict ordering sequence when it runs Start or Stop scripts. The default is **serial**, the strict ordering sequence.

[**--recoveryScript** <*script*>]
A script that runs after a monitor probe has failed. The script attempts to restore the device. The script pathname can be up to 512 characters long.

[**--recoveryTimeout** <*seconds*>]
The amount of time to wait for the Recovery script to complete.

[**--startScript** <*script*>]
When HP Clustered File System selects the active server for a monitored device, the Start script runs on that server. The script pathname can be up to 512 characters long.

[**--stopScript** <*script*>]
A script that runs on all other servers configured for the monitor to ensure that the device is not active on those servers. The script pathname can be up to 512 characters long.

[**--startTimeout** <*seconds*>]
The amount of time to wait for the Start script to complete.

[**--stopTimeout** <*seconds*>]
The amount of time to wait for the Stop script to complete.

[**--scriptSeverity consider**|**ignore**]
Whether HP Clustered File System takes device monitor events (such as a failure or timeout of a Start or Stop script) into consideration when it makes failover decisions. The default is **consider**.

[**--probe single**|**multiple**]
For custom monitors only, whether the monitor probe occurs on only one server or on all of the configured servers.

[**--activity single**|**multiple**]
For custom monitors only, whether the device monitor can be active on only one server or on all of the configured servers.

[**--vhosts** <*vhost1*>,<*vhost2*>,...]
The virtual hosts associated with the device monitor. These virtual hosts will fail over to another network interface when the monitored device fails. The default is all virtual hosts on the server(s) configured for the monitored device.

Modify (update) a device monitor:

mx device update [*arguments*] <*devicename*> ...

The arguments are the same as the **device add** command; however, the **--servers** argument is not required.

Enable a device monitor on a previously disabled server:

mx device enable <*devicename*> ...

Disable a device monitor on a server:

mx device disable <*devicename*> ...

Delete a device monitor:

mx device delete <*devicename*> ...

Clear a device event:

mx device clear <*devicename*> <*server*> ...

Display the status of a device monitor:

mx device status [*arguments*] [<*devicename*> ...]

The arguments are:

[**--up**|**--down**]
[**--enabled**|**--disabled**]
[**--primary**|**--backup**]
[**--active**|**--inactive**]

With no arguments, the command displays the status of all device monitors.

Dump device monitor configuration to stdout:

mx device dump

mx disk – disk commands

Use the following commands to import SAN disks into a cluster, to remove them from a cluster, or to display status information.

Import a disk into the cluster:

mx disk import <*uuid*> ...

Use the **disk status** command to determine the *uuid* for the disk.

This command does not display an error message if the import fails. To verify that the disk was imported, use the **mx disk status --imported** command.

Display information about disks:

mx disk status [**--imported**]

With no arguments, this command displays the uuid, the size, and a vendor string for each unimported disk in the SAN. To see this information for imported disks, include the **--imported** argument.

Remove a disk from the cluster:

mx disk deport <*uuid*> ...

Use the **disk status --imported** command to determine the *uuid* for the disk.

Dump disk configuration to stdout:

mx disk dump

mx dynvolume – dynamic volume commands

Use the following commands to create, recreate, extend, or destroy dynamic volumes, to display information about dynamic volumes, and to convert basic volumes to dynamic volumes.

Create a dynamic volume:

mx dynvolume create [**--stripesize** <*4KB-64MB*>] <*subDeviceName,subDeviceName,...*>

List subdevices available for use in a dynamic volume:

mx dynvolume showcreateopt

Display properties for a dynamic volume:

mx dynvolume properties <*volumeName*>

Show all dynamic volumes:

mx dynvolume list

Extend a dynamic volume:

mx dynvolume extend <*volumeName*> <*subDeviceName, subDeviceName,...*>

Destroy a dynamic volume:

mx dynvolume destroy <*volumeName*>

A filesystem cannot be mounted on the volume to be destroyed.

Convert a basic device to a dynamic volume:

mx dynvolume convert <*filesystem*>

Dump dynamic volume configuration to stdout:

mx dynvolume dump

mx exportgroup – Export Group commands

These commands are used to manage Export Groups.

Add an Export Group:

mx exportgroup add [**--exports** <*exports_file*>] [*other-arguments*] [**--vnfs** <*vnfs1*>,<*vnfs2*>,...] <*exportgroup_name*>

The **--exports** option specifies a file containing export records that will be imported into the Export Group. The **--vnfs** option specifies Virtual NFS Services that should be associated with this Export Group.

The other arguments are:

[**--timeout** <*seconds*>]
The maximum amount of time to wait for the monitor probe to complete. The default is 15 seconds.

[**--frequency** <*seconds*>]
The interval at which the high-availability monitor probes the NFS service. The default is 30 seconds.

[**--probeSeverity nofailover**|**autorecover**|**noautorecover**]
The failover behavior for the high-availability monitor. **nofailover** prevents failover of Virtual NFS Services when the probe fails. **autorecover** fails over the Virtual NFS Services, and when the NFS service is restored on the original node, fails the Virtual NFS Services back to the original network interfaces. **noautorecover** fails over the Virtual NFS Services but does not fail them back after the NFS service restored. The default is **autorecover**.

[**--ordering serial**|**parallel**]
Whether a strict ordering sequence should be enforced when Start or Stop scripts are run. The default is **serial**, the strict ordering sequence.

[**--recoveryScript** <*script*>]
A script that runs after a monitor probe has failed. The script attempts to restore the NFS service. The script pathname can be up to 512 characters long.

[**--recoveryTimeout** <*seconds*>]
The amount of time to wait for the Recovery script to complete.

[**--startScript** <*script*>]
A script that runs when the NFS service become active on a server. The script pathname can be up to 512 characters long.

[**--stopScript** <*script*>]
A script that runs on all other servers configured for the high-availability monitor to ensure that the NFS service is not active on those servers. The script pathname can be up to 512 characters long.

[**--startTimeout** <*seconds*>]
The amount of time to wait for the Start script to complete.

[**--stopTimeout** <*seconds*>]
The amount of time to wait for the Stop script to complete.

[**--scriptSeverity consider**|**ignore**]
Whether HP Clustered File System should take monitor events (such as a failure or timeout of a Start or Stop script) into consideration when it makes failover decisions. The default is **consider**.

[**--novalidate**]
Do not validate the export records.

Update an Export Group:

mx exportgroup update [**--exports** <*exports_file*>] [*other-arguments*]
--vnfs <*vnfs1*>,<*vnfs2*>,... <*exportgroup_name*>

The arguments are the same as the **exportgroup add** command.

View Export Groups:

mx exportgroup status [**--up**|**--down**] [**--enabled**|**--disabled**] [**--primary**|**--backup**] [**--active**|**--inactive**] [<*exportgroup_name*> ...]

Enable an Export Group:

mx exportgroup enable <*exportgroup_name*> [**ALL_SERVERS**]|[<*server*> ...]

Disable an Export Group:

mx exportgroup disable <*exportgroup_name*> [**ALL_SERVERS**]|[<*server*> ...]

Delete an Export Group:

mx exportgroup delete <*exportgroup_name*> ...

Clear an error associated with an Export Group

mx exportgroup clear <*exportgroup_name*> <*server*> ...

Dump Export Group configuration to stdout:

mx exportgroup dump

mx fs – filesystem commands

Use the following commands to create, mount, or unmount PSFS filesystems, and to display status information.

Create a PSFS filesystem:

mx fs create [**--size** <*KB*>] [**--options** <*options*>] <*filesystem*> <*storageExtent*>

In this release, a storage extent is a basic or dynamic volume. To locate an available storage extent, use the **fs showcreateopt** command. The **--options** argument is currently unused. The <*filesystem*> label can include up to 32 characters.

Display information about storage extents:

mx fs showcreateopt

Mount a PSFS filesystem on the specified servers:

mx fs mount [**--persist**] [**--activate**] [**--options** <*option,option,...*>] **--path** <*path*> <*filesystem*> <*server*> ...

The **--persist** argument causes the filesystem to be mounted automatically whenever the server is rebooted. The **--activate** argument mounts the filesystem now. The **--path** argument specifies the directory mountpoint for the filesystem; this mountpoint must already exist. <*filesystem*> is the label given to the filesystem when it was created.

The mount options are as follows:

RW/RO
: Mount the filesystem read-write or read-only. RW is the default.

ASYNC/ SYNC
: ASYNC, the default, allows either asychronous or synchronous I/O to the filesystem. SYNC allows only synchronous I/O.

DEV/ NODEV
: Interpret (or do not interpret) character or block special devices on the filesystem. DEV is the default.

EXEC/ NOEXEC
: Permit (or do not permit) the execution of binaries on the mounted filesystem. EXEC is the default. NOEXEC can be used on a system that has filesystems containing binaries for other architectures.

SUID/ NOSUID
: Allow (or do not allow) set-user-id bits and set-group-id bits to take effect. SUID is the default.

SHARED/ EXCLUSIVE
: Either allow all servers having physical access to the filesystem to mount it or allow only one server. SHARED is the default.

ORDERED/ UNORDERED
: With the ORDERED option, if a metadata operation will allocate user blocks, the user blocks are written to the filesystem before the metadata is written. With the UNORDERED option, the writes can occur in either order. ORDERED is the default.

Unmount a filesystem:

```
mx fs unmount [--persistent] [--active] <filesystem>
<server> ...
```

The **--persistent** argument removes the persistent status from the filesystem mount. The **--active** argument unmounts the filesystem now.

Display status information:

```
mx fs status [--mounted|--unmounted] [--persistent]
[--verbose] [--standard|--snapshot]
```

The **--mounted** argument displays only mounted filesystems; **--unmounted** displays only unmounted filesystems. The **--persistent** argument displays only those filesystems with persistent mounts. The **--verbose** option displays the FS type (always PSFS), the size of the filesystem in KB, and the UUID of the parent disk. The **--standard** argument shows only standard filesystems; the **--snapshot** argument shows only snapshots.

Dump filesystem configuration to stdout:

```
mx fs dump
```

mx matrix – cluster commands

Use these commands to display status and alert information, to dump cluster configuration information to *stdout*, or to destroy a cluster.

Display status information:

```
mx matrix status
```

The current alert messages appear at the end of the output.

Display alerts:

```
mx alert status [--severity <level>]
```

The **alert status** command lists the current alert messages. (These messages are also displayed on the Management Console.) The **--severity** option filters the alerts according to the specified alert level. There are four levels: error, fault, warning, and info.

Dump or restore cluster configuration information:

The following command dumps configuration information to *stdout*. The information includes the configuration of servers, network interfaces, virtual hosts, service and device monitors, notifiers, disks, and filesystems. It also includes sleep and alerts.

The dumped information does not include configuration information for the following: HP Clustered File System license, secret, cluster password, membership partitions, fencing configuration, FibreChannel switches.

```
mx matrix dump
```

You can also save the configuration in a file for backup purposes:

```
mx matrix dump > <backup_file>
```

To restore the configuration, use this command:

```
mx --continue --file <backup_file>
```

Destroy the cluster:

```
mx matrix destroy
```

This command attempts to remove your cluster configuration, including unmounting PSFS filesystems. The command may fail, based on the resources in use throughout the cluster.

The output from **matrix dump**, when run prior to the **matrix destroy** command, can be useful in restoring a cluster configuration after it has been destroyed; however, you may need to perform the reconfiguration manually.

mx netif – network interface commands

Use the following commands to manage network interfaces or to display their status.

Add a network interface:

```
mx netif add [--adminTraffic allow|discourage] --server
<server> --netmask <interface_netmask> <interface_ip>
```

--adminTraffic specifies whether the network interface should be available for HP Clustered File System administrative traffic. **allow** is the default.

Update a network interface:

```
mx netif update [--adminTraffic allow|discourage] --netmask
<interface_netmask> <interface_ip>
```

The network interface must be down.

Remove a network interface:

mx netif delete *<interface_ip>*

Enable a network interface for virtual hosting:

mx netif enable *<interface_ip>* **...**

Disable a network interface for virtual hosting:

mx netif disable *<interface_ip>* **...**

Make a network available for use administrative traffic:

mx netif admin *<interface_ip>* **...**

Do not use the network for administrative traffic:

mx netif noadmin *<interface_ip>* **...**

Display status for network interfaces:

mx netif status [**--up**|**--down**] [**--enabled**|**--disabled**] [**--admin**|**--noadmin**] [**--active**] [*<interface_ip>* **...**]

--up|**--down** selects network interfaces that are either operational or down. **--enabled**|**--disabled** selects interfaces that are either enabled or disabled for virtual hosting. **--admin**|**--noadmin** selects interfaces that either allow or discourage administrative traffic. **--active** selects interfaces that are currently handling administrative traffic.

mx notifier – notifier commands

Use the following commands to manage notifiers or to display their status.

Add a notifier:

mx notifier add --script *<script>* [**--event** <{**STATE**|**INFO**|**WARN**|**ERROR**},...>] [**--entity** <{**SERVERS**|**IFACES**|**VHOSTS**|**SERVICEMONITORS**|**DEVICEMONITORS**|**FILESYSTEMS**},...>] *<notifier>*

The *<notifier>* name can include up to 32 characters.

Update a notifier:

mx notifier update [**--script** *<script>*] [**--event** <{**STATE**|**INFO**|**WARN**|**ERROR**},...>] [**--entity** {**SERVERS**|**IFACES**|**VHOSTS**|**SERVICEMONITORS**|**DEVICEMONITORS**|**FILESYSTEMS**},...] *<notifier>*

Delete a notifier:

mx notifier delete *<notifier> ...*

Enable a notifier:

mx notifier enable *<notifier> ...*

Disable a notifier:

mx notifier disable *<notifier> ...*

Display status for notifiers:

mx notifier status [**--enabled**|**--disabled**] [*<notifier> ...*]

Test a notifier:

mx notifier test *<notifier> ...*

Dump the notifier configuration to stdout:

mx notifier dump [*<notifier> ...*]

mx server – server commands

Use the following commands to configure servers or to display their status.

Add a server to the cluster:

mx server add [**--serverSeverity autorecover**|**noautorecover**] *<server> ...*

The **--serverSeverity** argument specifies whether virtual hosts will automatically fail back to the original server after that server has recovered from a complete system outage. The default is **autorecover**.

Update one or more servers in a cluster:

mx server update [**--serverSeverity autorecover**|**noautorecover**] *<server> ...*

Delete one or more servers from a cluster:

mx server delete *<server> ...*

Disable one or more servers:

mx server disable *<server> ...*

Enable one or more previously disabled servers:

mx server enable <*server*> ...

Display server status:

mx server status [**--enabled**|**--disabled**] [**--up**|**--down**] [<*server*> ...]

Dump the server configuration to stdout:

mx server dump

mx service – service monitor commands

Use the following commands to configure service monitors or to display their status.

Add a service monitor:

mx service add [*arguments*] <*vhost:port*> ...

The arguments are:

[**--timeout** <*seconds*>]
The maximum amount of time to wait for a probe of the service to complete. For CUSTOM service monitors, the default is 60 seconds. For all other service monitors, the default is five seconds.

[**--frequency** <*seconds*>]

The interval of time at which the monitor probes the service. For CUSTOM service monitors, the default is 60 seconds. For all other service monitors, the default is 30 seconds.

[**--type** <*type*>]
The type of service monitor (FTP, HTTP, IMAP4, NFS, NIS, NNTP, POP3, SMTP, TCP, CUSTOM).

[**--probeSeverity nofailover**|**autorecover**|**noautorecover**]
The failover behavior for the service monitor. **nofailover** prevents failover of the virtual host when the monitored service fails. **autorecover** fails over the virtual host and, when the service is restored, fails the virtual host back to the original network interface. **noautorecover** fails over the virtual host but does not fail it back after the service is restored. **autorecover** is the default.

[**--parameters** <*parameters*>]
For an HTTP monitor, the parameter is the URL for the service. For an NFS monitor, the parameters are **proto** (either udp or tcp), **rpcname** (typically nfs), and **rpcvers** (either 2 or 3). For an NIS monitor, the parameters are **proto** (either udp or tcp), **rpcname** (typically ypserv), **rpcvers** (either 1 or 2), and **domain** (the NIS domain to be monitored). For a custom monitor, the parameter is a probe script.

[**--ordering serial**|**parallel**]
Whether HP Clustered File System enforces a strict ordering sequence when it runs Start or Stop scripts. The default is **serial**, the strict ordering sequence.

[**--recoveryScript** <*script*>]
Runs after a monitor probe failure is detected, in an attempt to restore the service.

[**--recoveryTimeout** <*seconds*>]
The amount of time to wait for the script to complete.

[**--startScript** <*script*>]
When HP Clustered File System selects the active server for a monitored service, the Start script runs on that server.

[**--stopScript** <*script*>]
A script that runs on all other servers configured for the service monitor to ensure that the service is not active on those servers.

[**--startTimeout** <*seconds*>]
The amount of time to wait for the Start script to complete.

[**--stopTimeout** <*seconds*>]
The amount of time to wait for the Stop script to complete.

[**--scriptSeverity consider**|**ignore**]
Whether HP Clustered File System takes service monitor events (such as a failure or timeout of a Start or Stop script) into consideration when it makes failover decisions. The default is **consider**.

[**--priority** <*priority*>]

The priority of the service monitor in relation to other service monitors. *priority* is a natural number; 0 is the highest priority.
HP Clustered File System uses the priority when failing over the virtual host associated with the monitor. If multiple failures have occurred, all of the services associated with the virtual host may not be available on one server. In this case, HP Clustered File System will fail over to a network interface on a server running the highest priority service. The default priority is 0.

Modify (update) an existing service monitor:

mx service update [*arguments*] <*vhost:port*> ...

The arguments are the same as **service add**.

Enable a service monitor:

mx service enable <*vhost:port*> <*server*> ...

Disable a service monitor:

mx service disable <*vhost:port*> <*server*> ...

Delete a service monitor:

mx service delete <*vhost:port*>

Clear a monitor event:

mx service clear <*vhost:port*> <*server*> ...

Display the status of a service monitor:

mx service status [*arguments*] [<*vhost:port*>]

The arguments are:

[**--up**|**--down**]
[**--primary**|**--backup**]
[**--enabled**|**--disabled**]
[**--active**|**--inactive**]

With no arguments, the command displays the status of all service monitors.

Dump the service monitor configuration to stdout:

mx service dump

mx snapshot – snapshot commands

Use the following commands to create or destroy hardware snapshots.

Show snapshot options for the storage array associated with a volume:

mx snapshot showcreateopt *<volume>*

Create a snapshot:

mx snapshot create [**--terse**] [*<options>*] *<volume>*

The **--terse** option causes only the name of the snapshot volume to be printed on success.

Destroy a snapshot:

mx snapshot destroy *<volume>*

mx vhost – virtual host commands

Use the following commands to configure virtual hosts or to display their status.

Add a new virtual host:

mx vhost add [**--policy autofailback**|**nofailback**] *<vhost>* *<network_interface>* ...

The **--policy** option specifies the failback action that the virtual host will take following failover to a backup node. **autofailback** will fail back when a higher-ranked node provides equal or greater services than the backup node; **nofailback** fails back only if a higher-ranked node provides greater services than the backup node. **autofailback** is the default.

<vhost> can be either the hostname or the IP address for the virtual host. If you specify the hostname, it can include up to 32 characters.

The order in which you specify the network interfaces determines the ordering for failover. The first network interface is the primary interface; network traffic for the virtual host goes to the server providing this interface. The remaining network interfaces are backups. Each network interface must be located on a different server.

Modify (update) an existing virtual host:

mx vhost update [**--policy autofailback**|**nofailback**] <*vhost*> <*network_interface*> ...

Use this option to change the policy or the ordering of the network interfaces for failover.

Delete a virtual host:

mx vhost delete <*vhost*>

Display the status of a virtual host:

mx vhost status [*arguments*] [<*vhost*> ...]

The arguments are:

[**--primary**|**--backup**]
[**--active**|**--inactive**]
[**--up**|**--down**]
[**--enabled**|**--disabled**]

With no arguments, the command displays the status of all virtual hosts.

Dump the virtual host configuration to stdout:

mx vhost dump

mx vnfs – Virtual NFS Service commands

These commands are used to manage Virtual NFS Services.

Add a Virtual NFS Service

mx vnfs add [**--exportgroup** <*exportgroup_name*>|**NONE**]
[**--policy autofailback**|**nofailback**] <*vnfs*> <*nework_interface*> ...

The **--exportgroup** option specifies the export group to be associated with this Virtual NFS Service. The **--policy** option specifies the failback action that the Virtual NFS Service will take following failover to a backup node. **autofailback** will fail back when a higher-ranked node provides equal or greater services than the backup node; **nofailback** fails back only if a higher-ranked node provides greater services than the backup node. **autofailback** is the default.

vnfs is the hostname or IP address of the Virtual NFS Service.

The order in which you specify the network interfaces determines the ordering for failover. The first network interface is the primary interface; network traffic for the virtual host goes to the node providing this interface. The remaining network interfaces are backups. Each network interface must be located on a different node.

Update a Virtual NFS Service:

```
mx vnfs update [--exportgroup <exportgroup_name>|NONE]
[--policy autofailback|nofailback] <vnfs> <netif> ...
```

Enable a Virtual NFS Service:

```
mx vnfs enable <vnfs>
```

Disable a Virtual NFS Service:

```
mx vnfs disable <vnfs>
```

Delete a Virtual NFS Service:

```
mx vnfs delete <vnfs> ...
```

View Virtual NFS Services:

```
mx vnfs status [--up|--down] [--enabled|--disabled]
[--primary|--backup] [--active|--inactive] [<vnfs> ...]
```

Move a Virtual NFS Service:

```
mx vnfs move <vnfs> <networkinterface> ...
```

Dump the Virtual NFS Service configuration to stdout:

```
mx vnfs dump
```

sleep command

The **sleep** command is useful when executing **mx** commands from a batch file. It has the following syntax:

```
sleep [seconds]
```

The **sleep** command can be inserted between **mx** commands in the batch file and causes **mx** to pause the specified number of seconds before executing the next command.

Index

V